Artificial Intelligence Review **22:** 85–126, 2004.
© 2004 *Kluwer Academic Publishers. Printed in the Netherlands.*

A Survey of Outlier Detection Methodologies

VICTORIA J. HODGE & JIM AUSTIN
Department of Computer Science, University of York, York, YO10 5DD UK
(E-mail: {vicky, austin}@cs.york.ac.uk)

Abstract. Outlier detection has been used for centuries to detect and, where appropriate, remove anomalous observations from data. Outliers arise due to mechanical faults, changes in system behaviour, fraudulent behaviour, human error, instrument error or simply through natural deviations in populations. Their detection can identify system faults and fraud before they escalate with potentially catastrophic consequences. It can identify errors and remove their contaminating effect on the data set and as such to purify the data for processing. The original outlier detection methods were arbitrary but now, principled and systematic techniques are used, drawn from the full gamut of Computer Science and Statistics. In this paper, we introduce a survey of contemporary techniques for outlier detection. We identify their respective motivations and distinguish their advantages and disadvantages in a comparative review.

Keywords: anomaly, detection, deviation, noise, novelty, outlier, recognition

1. Introduction

Outlier detection encompasses aspects of a broad spectrum of techniques. Many techniques employed for detecting outliers are fundamentally identical but with different names chosen by the authors. For example, authors describe their various approaches as outlier detection, novelty detection, anomaly detection, noise detection, deviation detection or exception mining. In this paper, we have chosen to call the technique outlier detection although we also use novelty detection where we feel appropriate but we incorporate approaches from all five categories named above. Additionally, authors have proposed many definitions for an outlier with seemingly no universally accepted definition. We will take the definition of Grubbs (1969) and quoted in Barnett and Lewis (1994).

> An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs.

A further outlier definition from Barnett and Lewis (1994) is:

> An observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data.

In Figure 2, there are five outlier points labelled V, W, X, Y and Z which are clearly isolated and inconsistent with the main cluster of points. The data in the figures in this survey paper is adapted from the Wine data set (Blake and Merz, 1998).

John (1995) states that an outlier may also be 'surprising veridical data', a point belonging to class A but actually situated inside class B so the true (veridical) classification of the point is surprising to the observer. Aggarwal and Yu (2001) note that outliers may be considered as noise points lying outside a set of defined clusters or alternatively outliers may be defined as the points that lie outside of the set of clusters but are also separated from the noise. These outliers behave differently from the norm. In this paper, we focus on the two definitions quoted from Barnett and Lewis (1994) above and do not consider the dual class-membership problem or separating noise and outliers.

Outlier detection is a critical task in many safety critical environments as the outlier indicates abnormal running conditions from which significant performance degradation may well result, such as an aircraft engine rotation defect or a flow problem in a pipeline. An outlier can denote an anomalous object in an image such as a land mine. An outlier may pinpoint an intruder inside a system with malicious intentions so rapid detection is essential. Outlier detection can detect a fault on a factory production line by constantly monitoring specific features of the products and comparing the real-time data with either the features of normal products or those for faults. It is imperative in tasks such as credit card usage monitoring or mobile phone monitoring to detect a sudden change in the usage pattern which may indicate fraudulent usage such as stolen card or stolen phone airtime. Outlier detection accomplishes this by analysing and comparing the time series of usage statistics. For application processing, such as loan application processing or social security benefit payments, an outlier detection system can detect any anomalies in the application before approval or payment. Outlier detection can additionally monitor the circumstances of a benefit claimant over time to ensure the payment has not slipped into fraud. Equity or commodity traders can use outlier detection methods to monitor individual shares or markets and detect novel trends which may indicate buying or selling opportunities. A news delivery system can detect changing news stories and ensure the supplier is first with the breaking news. In a database, outliers may indicate fraudulent cases or they may just denote an error by the entry clerk or a misinterpretation of a missing value code, either way detection of the anomaly is vital for data base consistency and integrity.

A more exhaustive list of applications that utilise outlier detection is:

- Fraud detection – detecting fraudulent applications for credit cards, state benefits or detecting fraudulent usage of credit cards or mobile phones.
- Loan application processing – to detect fraudulent applications or potentially problematical customers.
- Intrusion detection – detecting unauthorised access in computer networks.
- Activity monitoring – detecting mobile phone fraud by monitoring phone activity or suspicious trades in the equity markets.
- Network performance – monitoring the performance of computer networks, for example to detect network bottlenecks.
- Fault diagnosis – monitoring processes to detect faults in motors, generators, pipelines or space instruments on space shuttles for example.
- Structural defect detection – monitoring manufacturing lines to detect faulty production runs for example cracked beams.
- Satellite image analysis – identifying novel features or misclassified features.
- Detecting novelties in images – for robot neotaxis or surveillance systems.
- Motion segmentation – detecting image features moving independently of the background.
- Time-series monitoring – monitoring safety critical applications such as drilling or high-speed milling.
- Medical condition monitoring – such as heart-rate monitors.
- Pharmaceutical research – identifying novel molecular structures.
- Detecting novelty in text – to detect the onset of news stories, for topic detection and tracking or for traders to pinpoint equity, commodities, FX trading stories, outperforming or under performing commodities.
- Detecting unexpected entries in databases – for data mining to detect errors, frauds or valid but unexpected entries.
- Detecting mislabelled data in a training data set.

Outliers arise because of human error, instrument error, natural deviations in populations, fraudulent behaviour, changes in behaviour of systems or faults in systems. How the outlier detection system deals with the outlier depends on the application area. If the outlier indicates a typographical error by an entry clerk then the entry clerk can be notified and simply correct the error so the outlier will be restored to a normal record. An outlier resulting from an instrument reading error can simply be expunged. A survey of human population features may include anomalies such as a handful of very tall people. Here the anomaly is purely natural, although the reading may be worth flagging for verification to ensure no errors, it should be included in the classification once it is verified. A system should use a classification algorithm that is robust to outliers to model data with naturally occurring outlier points. An outlier in a safety critical environment, a fraud detection system, an image analysis system or an intrusion monitoring system must be detected immediately (in real-time) and a suitable alarm sounded to alert the system administrator to the problem. Once the situation has been handled, this anomalous reading may be stored separately for comparison with any new fraud cases but would probably not be stored with the main system data as these techniques tend to model normality and use this to detect anomalies.

There are three fundamental approaches to the problem of outlier detection:

1. **Type 1** – Determine the outliers with no prior knowledge of the data. This is essentially a learning approach analogous to *unsupervised clustering*. The approach processes the data as a static distribution, pinpoints the most remote points, and flags them as potential outliers. Type 1 assumes that errors or faults are separated from the 'normal' data and will thus appear as outliers. In Figure 3, points V, W, X, Y and Z are the remote points separated from the main cluster and would be flagged as possible outliers. We note that the main cluster may be subdivided if necessary into more than one cluster to allow both classification and outlier detection as with Figure 4. The approach is predominantly retrospective and is analogous to a batch-processing system. It requires that all data be available before processing and that the data is static. However, once the system possesses a sufficiently large database with good coverage, then it can compare new items with the existing data. There are two sub-techniques commonly employed, *diagnosis* and *accommodation* (Rousseeuw and Leroy, 1996). An outlier *diagnostic* approach highlights the potential outlying points. Once detected, the system may remove these outlier points from future processing of the data distribution. Many

diagnostic approaches iteratively prune the outliers and fit their system model to the remaining data until no more outliers are detected. An alternative methodology is *accommodation* that incorporates the outliers into the distribution model generated and employs a robust classification method. These robust approaches can withstand outliers in the data and generally induce a boundary of normality around the majority of the data which thus represents normal behaviour. In contrast, non-robust classifier methods produce representations which are skewed when outliers are left in. Non-robust methods are best suited when there are only a few outliers in the data set (as in Figure 3) as they are computationally cheaper than the robust methods but a robust method must be used if there are a large number of outliers to prevent this distortion. Torr and Murray (1993) use a cheap Least Squares algorithm if there are only a few outliers but switch to a more expensive but robust algorithm for higher frequencies of outliers.

2. **Type 2** – Model both normality and abnormality. This approach is analogous to *supervised classification* and requires pre-labelled data, tagged as normal or abnormal. In Figure 4, there are three classes of normal data with pre-labelled outliers in isolated areas. The entire area outside the normal class represents the outlier class. The normal points could be classified as a single class or subdivided into the three distinct classes according to the requirements of the system to provide a simple normal/abnormal classification or to provide an abnormal and 3-classes of normally classifier.

   Classifiers are best suited to static data as the classification needs to be rebuilt from first principles if the data distribution shifts unless the system uses an incremental classifier such as an evolutionary neural network. We describe one such approach later which uses a grow when required (Marsland, 2001) network. A type 2 approach can be used for on-line classification, where the classifier learns the classification model and then classifies new exemplars as and when required against the learned model. If the new exemplar lies in a region of normality it is classified as normal, otherwise it is flagged as an outlier. Classification algorithms require a good spread of both normal and abnormal data, i.e., the data should cover the entire distribution to allow generalisation by the classifier. New exemplars may then be classified correctly as classification is limited to a 'known' distribution and a new exemplar derived from a previously unseen region of the distribution may not be classified correctly unless the generalisation capabilities of the underlying classification algorithm are good.

3. **Type 3** - Model only normality or in a very few cases model abnormality (Japkowicz et al., 1995; Fawcett and Provost 1999). Authors generally name this technique novelty detection or novelty recognition. It is analogous to a *semi-supervised recognition or detection* task and can be considered semi-supervised as the normal class is taught but the algorithm learns to recognise abnormality. The approach needs pre-classified data but only learns data marked normal. It is suitable for static or dynamic data as it only learns one class which provides the model of normality. It can learn the model incrementally as new data arrives, tuning the model to improve the fit as each new exemplar becomes available. It aims to define a boundary of normality.

   A type 3 system recognises a new exemplar as normal if it lies within the boundary and recognises the new exemplar as novel otherwise. In Figure 5, the novelty recogniser has learned the same data as shown in Figure 2 but only the normal class is learned and a boundary of normality induced. If points V, W, X, Y and Z from Figure 2 are compared to the novelty recogniser they will be labelled as abnormal as they lie outside the induced boundary. This boundary may be hard where a point lies wholly within or wholly outside the boundary or soft where the boundary is graduated depending on the underlying detection algorithm. A soft bounded algorithm can estimate the degree of 'outlierness'.

   It requires the full gamut of normality to be available for training to permit generalisation. However, it requires no abnormal data for training unlike type 2. Abnormal data is often difficult to obtain or expensive in many fault detection domains such as aircraft engine monitoring. It would be extremely costly to sabotage an aircraft engine just to obtain some abnormal running data. Another problem with type 2 is it cannot always handle outliers from unexpected regions, for example, in fraud detection a new method of fraud never previously encountered or previously unseen fault in a machine may not be handled correctly by the classifier unless generalisation is very good. In this method, as long as the new fraud lies outside the boundary of normality then the system will be correctly detect the fraud. If normality shifts then the normal class modelled by the system may be shifted by re-learning the data model or shifting the model if the underlying modelling technique permits such as evolutionary neural networks.

The outlier approaches described in this survey paper generally map data onto vectors. The vectors comprise numeric and symbolic attributes to represent continuous-valued, discrete (ordinal), categorical (unordered numeric), ordered symbolic or unordered symbolic data. The vectors may be monotype or multi-type. The statistical and neural network approaches typically require numeric monotype attributes and need to map symbolic data onto suitable numeric values[1] but the machine learning techniques described are able to accommodate multi-type vectors and symbolic attributes. The outliers are determined from the 'closeness' of vectors using some suitable distance metric. Different approaches work better for different types of data, for different numbers of vectors, for different numbers of attributes, according to the speed required and according to the accuracy required. The two fundamental considerations when selecting an appropriate methodology for an outlier detection system are:

- Selecting an algorithm which can accurately model the data distribution and accurately highlight outlying points for a clustering, classification or recognition type technique. The algorithm should also be scalable to the data sets to be processed.
- Selecting a suitable neighbourhood of interest for an outlier. The selection of the neighbourhood of interest is non-trivial. Many algorithms define boundaries around normality during processing and autonomously induce a threshold. However, these approaches are often parametric enforcing a specific distribution model or require user-specified parameters such as the number of clusters. Other techniques discussed below require user-defined parameters to define the size or density of neighbourhoods for outlier thresholding. The choice of neighbourhood whether user-defined or autonomously induced needs to be applicable for all density distributions likely to be encountered and can potentially include those with sharp density variations.

In the remainder of this paper, we categorise and analyse broad range of outlier detection methodologies. We pinpoint how each handles outliers and make recommendations for when each methodology is appropriate for clustering, classification and/or recognition. Barnett and Lewis (1994) and Rousseeuw and Leroy (1996) describe and analyse a broad range of statistical outlier techniques and Marsland (2001) analyses a wide range of neural methods. We have observed that outlier detection methods are derived from three fields of computing: statistics (proximity-based, parametric, non-parametric and semi-parametric), neural networks (supervised and unsupervised) and machine learning.

In the next four sections, we describe and analyse techniques from all three fields and a collection of hybrid techniques that utilise algorithms from multiple fields. The approaches described here encompass distance-based, set-based, density-based, depth-based, model-based and graph-based algorithms.

2. Statistical Models

Statistical approaches were the earliest algorithms used for outlier detection. Some of the earliest are applicable only for single dimensional data sets. In fact, many of the techniques described in both Barnett and Lewis (1994) and Rousseeuw and Leroy (1996) are single dimensional or at best univariate. One such single dimensional method is Grubbs' method (extreme studentized deviate) (Grubbs, 1969) which calculates a Z value as the difference between the mean value for the attribute and the query value divided by the standard deviation for the attribute where the mean and standard deviation are calculated from all attribute values including the query value. The Z value for the query is compared with a 1% or 5% significance level. The technique requires no user parameters as all parameters are derived directly from data. However, the technique is susceptible to the number of exemplars in the data set. The higher the number of records the more statistically representative the sample is likely to be.

Statistical models are generally suited to quantitative real-valued data sets or at the very least quantitative ordinal data distributions where the ordinal data can be transformed to suitable numerical values for statistical (numerical) processing. This limits their applicability and increases the processing time if complex data transformations are necessary before processing.

Probably one of the simplest statistical outlier detection techniques described here, Laurikkala et al. (2000) use informal box plots to pinpoint outliers in both univariate and multivariate data sets. This produces a graphical representation (see Figure 1 for an example box plot) and allows a human auditor to visually pinpoint the outlying points. It is analogous to a visual inspection of Figure 2. Their approach can handle real-valued, ordinal and categorical (no order) attributes. Box plots plot the lower extreme, lower quartile, median, upper quartile and upper extreme points. For univariate data, this is a simple 5-point plot, as in Figure 1. The outliers are the points beyond the lower and upper extreme values of the box plot, such as V, Y and Z

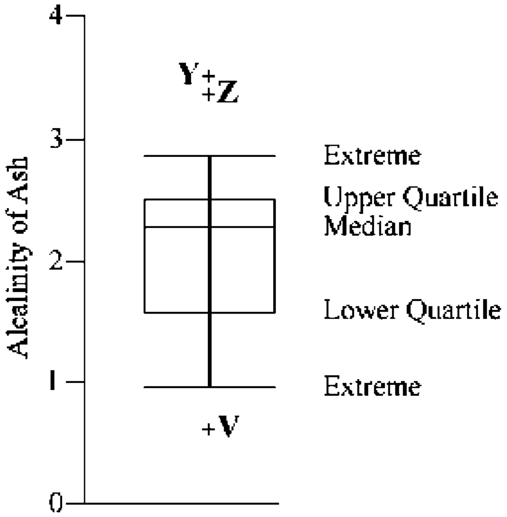

Figure 1. Shows a box plot of the *y*-axis values for the data in Figure 2 with the lower extreme, the lower quartile, median, upper quartile and upper extreme from the normal data and the three outliers V, Y and Z plotted. The points X and W from Figure 2 are not outliers with respect to the *y*-axis values.

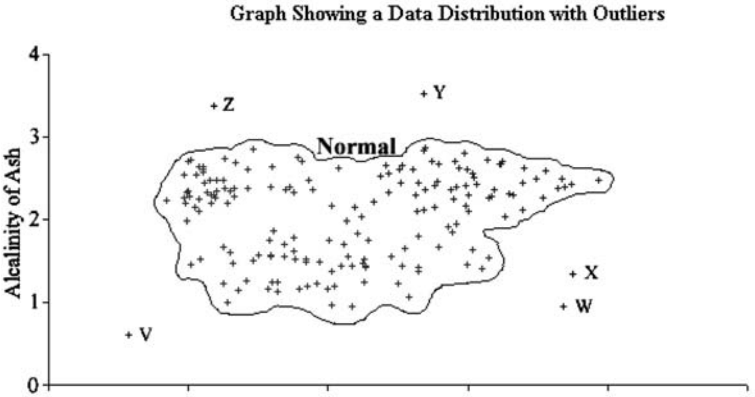

Figure 2. Shows a data distribution with 5 outliers (V, W, X, Y and Z). The data is adapted from the Wine data set (Blake and Merz, 1998).

in Figure 1. Laurikkala et al. suggest a heuristic of 1.5 × inter-quartile range beyond the upper and lower extremes for outliers but this would need to vary across different data sets. For multivariate data sets the authors note that there are no unambiguous total orderings but recommend using the reduced sub-ordering based on the generalised distance metric using the Mahalanobis distance measure (see equation (2)). The Mahalanobis distance measure includes the inter-attribute dependencies so the system can compare attribute combinations. The authors found the approach most accurate for multivariate data where a panel of experts agreed with the outliers detected by the system. For univariate data, outliers are more subjective and may be naturally occurring, for example the heights of adult humans, so there was generally more disagreement. Box plots make no assumptions about the data distribution model but are reliant on a human to note the extreme points plotted on the box plot.

Statistical models use different approaches to overcome the problem of increasing dimensionality which both increases the processing time and distorts the data distribution by spreading the convex hull. Some methods preselect key exemplars to reduce the processing time (Skalak, 1994; Datta and Kibler, 1995). As the dimensionality increases, the data points are spread through a larger volume and become less dense. This makes the convex hull harder to discern and is known as the 'Curse of Dimensionality'. The most efficient statistical techniques automatically focus on the salient attributes and are able to process the higher number of dimensions in tractable time. However, many techniques such as k-NN, neural networks, minimum volume ellipsoid or convex peeling described in this survey are susceptible to the Curse of Dimensionality. These approaches may utilise a preprocessing algorithm to preselect the salient attributes (Skalak and Rissland, 1990; Aha and Bankert, 1994; Skalak, 1994). These feature selection techniques essentially remove noise from the data distribution and focus the main cluster of normal data points while isolating the outliers as with Figure 2. Only a few attributes usually contribute to the deviation of an outlier case from a normal case. An alternative technique is to use an algorithm to project the data onto a lower dimensional subspace to compact the convex hull (Aggarwal and Yu, 2001) or use principal component analysis (Parra et al., 1996; Faloutsos et al., 1997).

2.1. *Proximity-based techniques*

Proximity-based techniques are simple to implement and make no prior assumptions about the data distribution model. They are suitable for both type 1 and type 2 outlier detection. However, they suffer exponential computational growth as they are founded on the calculation of the distances between all records. The computational complexity is directly proportional to both the dimensionality of the data m and the number of records n. Hence, methods such as k-nearest neighbour (also known as instance-based learning and described next) with $O(n^2m)$ runtime are not feasible for high dimensionality data sets unless the running time can be improved. There are various flavours of k-nearest neighbour (k-NN) algorithm for outlier detection but all calculate the nearest neighbours of a record using a suitable distance calculation metric such as Euclidean distance or Mahalanobis distance. Euclidean distance is given by equation (1):

$$\sqrt{\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{y}_i)^2} \tag{1}$$

and is simply the vector distance whereas the Mahalanobis distance given by equation (2):

$$\sqrt{(\mathbf{x} - \mu)^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x} - \mu)} \tag{2}$$

calculates the distance from a point to the centroid (μ) defined by correlated attributes given by the Covariance matrix (**C**). Mahalanobis distance is computationally expensive to calculate for large high dimensional data sets compared to the Euclidean distance as it requires a pass through the entire data set to identify the attribute correlations.

Ramaswamy et al. (2000) introduce an optimised k-NN to produce a ranked list of potential outliers. A point p is an outlier if no more than $n - 1$ other points in the data set have a higher D_m (distance to mth neighbour) where m is a user-specified parameter. In Figure 3, V is most isolated followed by X, W, Y then Z so the outlier rank would be V, X, W, Y, Z. This approach is susceptible to the computational growth as the entire distance matrix must be calculated for all points (*ALL k-NN*) so Ramaswamy et al. include techniques for speeding the k-NN algorithm such as partitioning the data into cells. If any cell and its directly adjacent neighbours contains more than k points, then the points in the cell are deemed to lie in a dense area of the distribution so the points contained are unlikely to be outliers. If the number of points lying in cells more than a pre-specified distance apart is less than k then all points in the cell are labelled as outliers. Hence, only a small number of cells not previously labelled need to be processed and only a relatively small number of distances need to be calculated for outlier detection. Authors have also improved the running speed of k-NN by creating an

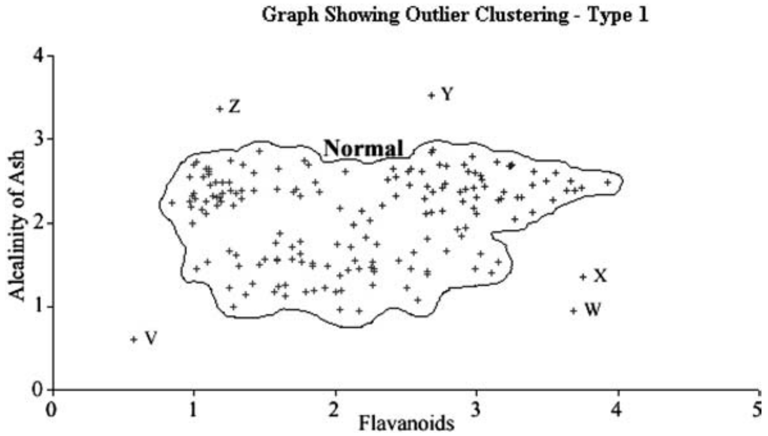

Figure 3. Shows a data distribution classified by type 1 – outlier clustering. The data is adapted from the Wine data set (Blake and Merz, 1998).

efficient index using a computationally efficient indexing structure (Ester et al., 1996) with linear running time.

Knorr and Ng (1998) introduce an efficient type 1 k-NN approach. If m of the k nearest neighbours (where $m < k$) lie within a specific distance threshold d then the exemplar is deemed to lie in a sufficiently dense region of the data distribution to be classified as normal. However, if there are less than m neighbours inside the distance threshold then the exemplar is an outlier. A very similar type 1 approach for identifying land mines from satellite ground images (Byers and Raftery, 1998) is to take the mth neighbour and find the distance D_m. If this distance is less than a threshold d then the exemplar lies in a sufficiently dense region of the data distribution and is classified as normal. However, if the distance is more than the threshold value then the exemplar must lie in a locally sparse area and is an outlier. This has reduced the number of data-specific parameters from Knorr and Ng's (1998) approach by one as we now have d and m but no k value. In Figure 3, points V, W, X, Y and Z are relatively distant from their neighbours and will have fewer than m neighbours within d and a high D_m so both Knorr and Ng and Byers and Raftery classify them as outliers. This is less susceptible to the computational growth than the *ALL k-NN* approach as only the k nearest neighbours need to be calculated for a new exemplar rather than the entire distance matrix for all points.

A type 2 classification k-NN method such as the majority voting approach (Wettschereck, 1994) requires a labelled data set with both normal and abnormal vectors classified. The k-nearest neighbours for the new exemplar are calculated and it is classified according to the majority classification of the nearest neighbours (Wettschereck, 1994). An extension incorporates the distance where the voting power of each nearest neighbour is attenuated according to its distance from the new item (Wettschereck, 1994) with the voting power systematically decreasing as the distance increases. Tang et al. (2002) introduce an type 1 outlier diagnostic which unifies weighted *k*-NN with a connectivity-based approach and calculates a weighted distance score rather than a weighted classification. It calculates the average chaining distance (path length) between a point *p* and its *k* neighbours. The early distances are assigned higher weights so if a point lies in a sparse region as points V, W, X, Y and Z in Figure 4 its nearest neighbours in the path will be relatively distant and the average chaining distance will be high. In contrast, Wettschereck requires a data set with good coverage for both normal and abnormal points. The distribution in Figure 4 would cause problems for voted *k*-NN as there are relatively few examples of outliers and their nearest neighbours, although distant, will in fact be normal points so they are classified as normal. Tang's underlying principle is to assimilate both density and isolation. A point can lie in a relatively sparse region of a distribution without being an outlier but a point in isolation is an outlier. However, the technique is computationally complex with a similar expected run-time to a full *k*-NN matrix calculation as it relies on calculating paths between all points and their *k* neighbours. Wettschereck's approach is less susceptible as only the *k* nearest neighbours are calculated relative to the single new item.

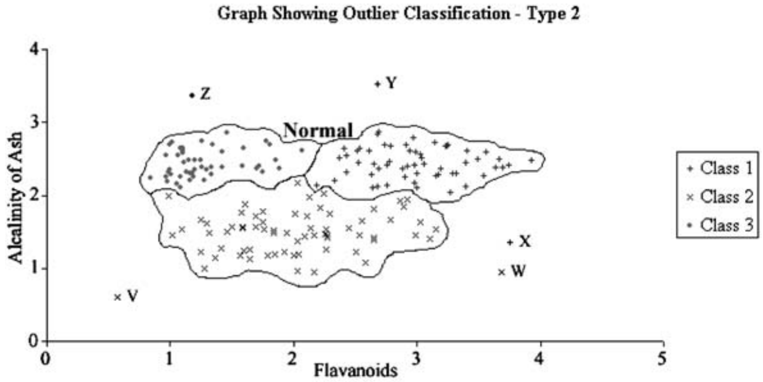

Figure 4. Shows a data distribution classified by type 2 – outlier classification. The data is adapted from the Wine data set (Blake and Merz, 1998).

Another technique for optimising *k*-NN is reducing the number of features. These feature subset selectors are applicable for most systems described in this survey, not just the *k*-NN techniques. Aha and Bankert (1994) describe a forward sequential selection feature selector which iteratively evaluates feature subsets adding one extra dimension per iteration until the extra dimension entails no improvement in performance. The feature selector demonstrates high accuracy when coupled with an instance-based classifier (nearest neighbour) but is computationally expensive due to the combinatorial problem of subset selection. Aggarwal and Yu (2001) employ lower dimensional projections of the data set and focus on key attributes. The method assumes that outliers are abnormally sparse in certain lower dimensional projections where the combination of attributes in the projection correlates to the attributes that are deviant. Aggarwal and Yu use an evolutionary search algorithm to determine the projections which has a faster running time than the conventional approach employed by Aha and Bankert.

Proximity based methods are also computationally susceptible to the number of instances in the data set as they necessitate the calculation of all vector distances. Datta and Kibler (1995) use a diagnostic prototype selection to reduce the storage requirements to a few seminal vectors. Skalak (1994) employs both feature selection and prototype selection where a large data set can be stored as a few lower dimensional prototype vectors. Noise and outliers will not be stored as prototypes so the method is robust. Limiting the number of prototypes prevents over-fitting just as pruning a decision tree can prevent over-fitting by limiting the number of leaves (prototypes) stored. However, prototyping must be applied carefully and selectively as it will increase the sparsity of the distribution and the density of the nearest neighbours. A majority voting *k*-NN technique such as Wettschereck (1994) will be less affected but an approach relying on the distances to the *n*th neighbour (Byers and Raftery, 1998) or counting the number of neighbours within specific distances (Knorr and Ng, 1998) will be strongly affected.

Prototyping is in many ways similar to the *k*-means and *k*-medoids described next but with prototyping a new instance is compared to the prototypes using conventional *k*-nearest neighbour whereas *k*-means and *k*-medoids prototypes have a kernel with a locally defined radius and the new instance is compared with the kernel boundaries. A prototype approach is also applicable for reducing the data set for neural networks and decision trees. This is particularly germane for node-based neural networks which require multiple passes through the data set to train so a reduced data set will entail less training steps.

Dragon Research (Allan et al., 1998) and Nairac (Nairac et al., 1999) (discussed in section 5) use k-means for novelty detection. Dragon perform on-line event detection to identify news stories concerning new events. Each of the k clusters provides a local model of the data. The algorithm represents each of the k clusters by a prototype vector with attribute values equivalent to the mean value across all points in that cluster. In Figure 4, if k is 3 then the algorithm would effectively circumscribe each class (1, 2 and 3) with a hyper-sphere and these three hyperspheres effectively represent and classify normality. k-Means requires the user to specify the value of k in advance. Determining the optimum number of clusters is a hard problem and necessitates running the k-means algorithm a number of times with different k values and selecting the best results for the particular data set. However, the k value is usually small compared with the number of records in the data set. Thus, the computational complexity for classification of new instances and the storage overhead are vastly reduced as new vectors need only be compared to k prototype vectors and only k prototype vectors need be stored unlike k-NN where all vectors need be stored and compared to each other.

k-Means initially chooses random cluster prototypes according to a user-defined selection process, the input data is applied iteratively and the algorithm identifies the best matching cluster and updates the cluster centre to reflect the new exemplar and minimise the sum-of-squares clustering function given by equation (3):

$$\sum_{j=1}^{K} \sum_{n \in S_j} \|\mathbf{x}^n - \mu_j\|^2 \tag{3}$$

where μ is the mean of the points ($\mathbf{x}^n$) in cluster S_j. Dragon use an adapted similarity metric that incorporates the word count from the news story, the distance to the clusters and the effect the insertion has on the prototype vector for the cluster. After training, each cluster has a radius which is the distance between the prototype and the most distant point lying in the cluster. This radius defines the bounds of normality and is local to each cluster rather than the global distance settings used in many approaches such as (Byers and Raftery, 1998; Knorr and Ng, 1998; Ramaswamy et al., 2000) k-NN approaches. A new exemplar is compared with the k-cluster model. If the point lies outside all clusters then it is an outlier.

A very similar partitional algorithm is the k-medoids algorithm or partition around medoids (PAM) which represents each cluster using an actual point and a radius rather than a prototype (average) point and a radius. Bolton and Hand (2001) use a k-medoids type approach they call peer group analysis for fraud detection. k-Medoids is robust to outliers as it does not use optimisation to solve the vector placement problem but rather uses actual data points to represent cluster centres. k-Medoids is less susceptible to local minima than standard k-means during training where k-means often converges to poor quality clusters. It is also data-order independent unlike standard k-means where the order of the input data affects the positioning of the cluster centres and Bradley et al. (1999) shows that k-medoids provides better class separation than k-means and hence is better suited to a novelty recognition task due to the improved separation capabilities. However, k-means outperforms k-medoids and can handle larger data sets more efficiently as k-medoids can require $O(n^2)$ running time per iteration whereas k-means is $O(n)$. Both approaches can generalise from a relatively small data set. Conversely, the classification accuracy of k-NN, least squares regression or Grubbs' method is susceptible to the number of exemplars in the data set like the kernel-based Parzen windows and the node-based supervised neural networks described later as they all model and analyse the density of the input distribution.

The data mining partitional algorithm CLARANS (Ng and Han, 1994), is an optimised derivative of the k-medoids algorithm and can handle outlier detection which is achieved as a by-product of the clustering process. It applies a random but bounded heuristic search to find an optimal clustering by only searching a random selection of cluster updates. It requires two user-specified parameters, the value of k and the number of cluster updates to randomly select. Rather than searching the entire data set for the optimal medoid it tests a pre-specified number of potential medoids and selects the first medoid it tests which improves the cluster quality. However, it still has $O(n^2k)$ running time so is only really applicable for small to medium size data sets.

Another proximity-based variant is the graph connectivity method. Shekhar et al. (2001) introduce an approach to traffic monitoring which examines the neighbourhoods of points but from a topologically connected rather than distance-based perspective. Shekhar detects traffic monitoring stations producing sensor readings which are inconsistent with stations in the immediately connected neighbourhood. A station is an outlier if the difference between its sensor value and the average sensor value of its topological neighbours differs by more than a threshold percentage from the mean difference between all nodes and their topological neighbours. This is analogous to calculating the average distance between each point i and its k neighbours avg D_i^k and then finding any points whose average distance avg D^k differs by more than a specified percentage. The technique only considers topologically connected neighbours so there is no prerequisite for specifying k and the number can vary locally depending on the number of connections. However, it is best suited to domains where a connected graph of nodes is available such as analysing traffic flow networks where the individual monitoring stations represent nodes in the connected network.

2.2 *Parametric methods*

Many of the methods we have just described do not scale well unless modifications and optimisations are made to the standard algorithm. Parametric methods allow the model to be evaluated very rapidly for new instances and are suitable for large data sets; the model grows only with model complexity not data size. However, they limit their applicability by enforcing a pre-selected distribution model to fit the data. If the user knows their data fits such a distribution model then these approaches are highly accurate but many data sets do not fit one particular model.

One such approach is minimum volume ellipsoid estimation (MVE) (Rousseeuw and Leroy, 1996) which fits the smallest permissible ellipsoid volume around the majority of the data distribution model (generally covering 50% of the data points). This represents the densely populated normal region shown in Figure 2 (with outliers shown) and Figure 5 (with outliers removed).

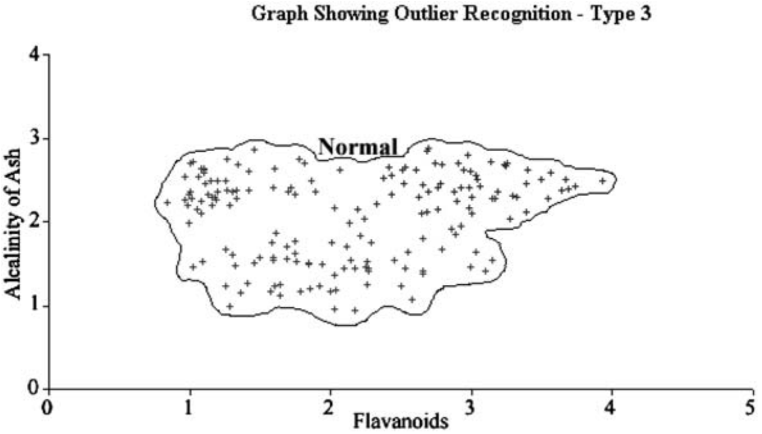

Figure 5. Shows a data distribution classified by type 3 – outlier recognition. The data is adapted from the Wine data set (Blake and Merz, 1998).

A similar approach, convex peeling peels away the records on the boundaries of the data distribution's convex hull (Rousseeuw and Leroy, 1996) and thus peels away the outliers. In contrast MVE maintains all points and defines a boundary around the majority of points. In convex peeling, each point is assigned a depth. The outliers will have the lowest depth thus placing them on the boundary of the convex hull and are shed from the distribution model. For example in Figure 2, V, W, X, Y and Z would each be assigned the lowest depth and shed during the first iteration. Peeling repeats the convex hull generation and peeling process on the remaining records until it has removed a pre-specified number of records. The technique is a type 1, unsupervised clustering outlier detector. Unfortunately, it is susceptible to peeling away $p+1$ points from the data distribution on each iteration and eroding the stock of normal points too rapidly (Rousseeuw and Leroy, 1996).

Both MVE and convex peeling are robust classifiers that fit boundaries around specific percentages of the data irrespective of the sparseness of the outlying regions and hence outlying data points do not skew the boundary. Both however, rely on a good spread of the data. Figure 2 has few outliers so an ellipsoid circumscribing 50% of the data would omit many normal points from the boundary of normality. Both MVE and convex peeling are only applicable for lower dimensional data sets (Barnett and Lewis, 1994) (usually three dimensional or less for convex peeling) as they suffer the curse of dimensionality where the convex hull is stretched as more dimensions are added and the surface becomes too difficult to discern.

Torr and Murray (1993) also peel away outlying points by iteratively pruning and re-fitting. They measure the effect of deleting points on the placement of the least squares standard regression line for a diagnostic outlier detector. The LS line is placed to minimise equation (4):

$$\sum_{i=1}^{n} (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2 \tag{4}$$

where $\hat{\mathbf{y}}_i$ is the estimated value. Torr and Murray repeatedly delete the single point with maximal influence (the point that causes the greatest deviation in the placement of the regression line) thus allowing the fitted model to stay away from the outliers. They refit the regression to the remaining data until there are no more outliers, i.e., the next point with maximal influence lies below a threshold value. Least squares regression is not robust as the outliers affect the placement of the regression line so it is best suited to outlier diagnostics where the outliers are removed

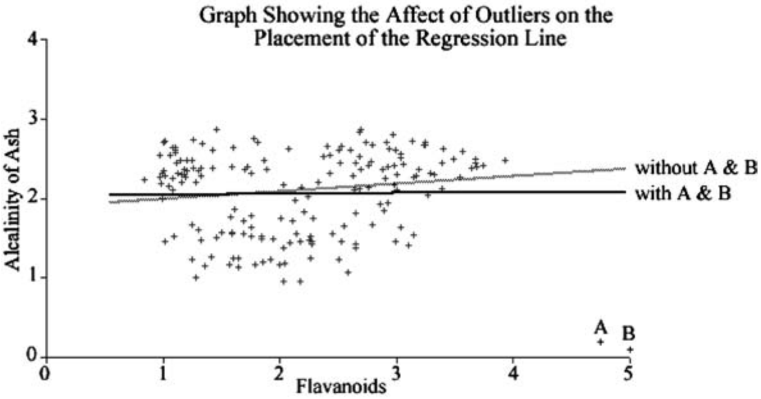

Figure 6. Shows a regression line fitted to the distribution shown in 2 with outliers A and B present (black line) and all outliers removed (grey line). The outliers affect the placement of the regression line.

from the next iteration. Figure 6 shows a least squares regression line fitted to a data distribution with the outliers A and B present and then again after point A and B have been removed. Although there are only two outliers, they have a considerable affect on the line placement.

Torr and Murray (1993) extend the technique for image segmentation. They use a computationally cheap non-robust least median of squares (LMS) regression if the number of outliers is small which minimises equation (5):

$$\mathrm{Median}_{i=1}^{n}(\mathbf{y}_i - \hat{\mathbf{y}}_i)^2 \tag{5}$$

or a computationally expensive robust random sampling algorithm if the number of outliers is high. LMS is able to accommodate more outliers than LS as it uses the median values. However, random sampling can accommodate larger numbers of outliers which eventually distort LMS. LMS has also been improved to produce the least trimmed squares approach (Rousseeuw and Leroy, 1996) which has faster convergence and minimises equation (6):

$$\sum_{i=1}^{h}((\mathbf{y}_i - \hat{\mathbf{y}}_i)^2)_{i:n} \tag{6}$$

where $(\mathbf{y}_i - \hat{\mathbf{y}}_i)^2)_{1:n} \leq \cdots \leq (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2)_{n:n}$ are the ordered square residuals. The summation function accommodates outliers in the distribution by fitting the regression to only the majority of the data rather than all of the data as in LMS. This region thus depicts normality and LTS highlights outliers as the points with large deviations from the majority.

MVE and convex peeling aim to compact the convex hull and circumscribe the data with a decision boundary but are only applicable for low dimensional data. Principal component analysis (PCA) (Parra et al., 1996; Faloutsos et al., 1997) in contrast, is suitable for higher dimensional data. It identifies correlated attributes in the data distribution and projects the data onto this lower dimensional subspace. PCA is an unsupervised classifier but is linear and incapable of outperforming the complex non-linear class boundaries identified by the support vector machine (see section 2.4) or neural methods described in section 3. PCA assumes that the subspaces determined by the principal components are compact and this limits its applicability particularly for sparse distributions. However, it is an ideal pre-processor to select a subset of attributes for methods which suffer the curse of dimensionality such as the multi-layer perceptron in section 3, proximity-based techniques or symplectic transformations described next. PCA identifies the principal component of greatest variance as each component has an associated eigenvalue whose magnitude corresponds to the variance of the points from the component vector. PCA retains the k principal components with greatest variance and discards all others to preserve maximum information and retain minimal redundancy.

Faloutsos et al. (1997) recommend retaining sufficient components so the sum of the eigenvalues of all retained components is at least 85% of the sum of all eigenvalues. They use the principal components to predict attribute values in records by finding the intersection between the given values for the record (i.e., excluding the omitted attribute) and the principal components. If the actual value for an attribute and the predicted value differ then the record is flagged as an outlier. Parra et al. (1996) have developed a type-3 motor fault detector system which applies PCA to the data and then applies a symplectic transformation to the first few principal components. The symplectic transformation may be used with non-linear data distributions. It maps the input data onto a Gaussian distribution, conserving the volume and separating the training data (normal data) from the outliers. This double transformation preserves information while removing redundancy, generates a density estimation of the data set and thus allows a circular contour of the density to act as the decision boundary.

Baker et al. (1999) employ one of the hierarchical approaches detailed in this survey. The other hierarchical approaches are the decision tree and cluster trees detailed in the machine learning section. Baker uses a parametric model-based approach for novelty detection in a news story monitor. A hierarchy allows the domain knowledge to be represented at various levels of abstraction so points can be compared for novelty at a fine-grained or less specific level. The hierarchical statistical algorithm induces a topic hierarchy from the word distributions of news stories using expectation maximisation (EM) to estimate the parameter settings followed by deterministic annealing (DA). DA constructs the hierarchy via maximum likelihood and information theory using a divisive clustering approach to split nodes into sub-nodes, starting from a single cluster, and build the hierarchy top–down. DA stochastically determines the node to split. The system detects novelty when new nodes are added to the hierarchy that represent documents that do not belong to any of the existing event clusters so new events are effectively described by their position in the hierarchy. When EM is used in conjunction with DA it avoids some of the initialisation dependence of EM but at the cost of computational efficiency. DA can avoid local minima which EM is susceptible to but it may produce sub-optimal results.

2.3. *Non-parametric methods*

Many statistical methods described in this section have data-specific parameters ranging from the *k* values of *k*-NN and *k*-means to distance thresholds for the proximity-based approaches to complex model parameters. Other techniques such as those based around convex hulls and regression and the PCA approaches assume the data follows a specific model. These all require *a priori* data knowledge. Such information is often not available or is expensive to compute. Many data sets simply do not follow one specific distribution model and are often randomly distributed. Hence, these approaches may be applicable for an outlier detector where all data is accumulated beforehand and may be pre-processed to determine parameter settings or for data where the distribution model is known. Non-parametric approaches, in contrast are more flexible and autonomous.

Dasgupta and Forrest (1996) introduce a non-parametric approach for novelty detection in machinery operation. The authors recognise novelty which contrasts to the other type 3 approaches we describe such as *k*-means (Nairac et al., 1999) or the ART neural approach (Caudell and Newman, 1993) in section 3 which recognise or classify the normal data space. The machinery operation produces a time-series of real-valued machinery measurements which Dasgupta and Forrest map onto binary vectors using quantisation (binning). The binary vector (string) effectively represents an encoding of the last n real-values from the time series. As the machinery is constantly monitored, new strings (binary vector windows) are generated to represent the current operating characteristics. Dasgupta and Forrest use a set of detectors where all detectors fail to match any strings defining normality (where two strings match if they are identical in a fixed number of contiguous positions (r)). If any detectors match a new string (a new time window of operating characteristics) then a novelty has been detected. The value of r affects the performance of the algorithm and the value must be selected carefully by empirical testing which inevitably slows processing. Each individual recogniser represents a subsection of the input distribution and compares the input. Dasgupta and Forrest's approach would effectively model Figure 5 by failing to match any point within the normal boundary but would match any point from outside.

2.4. *Semi-parametric methods*

Semi-parametric methods apply local kernel models rather than a single global distribution model. They aim to combine the speed and complexity growth advantage of parametric methods with the model flexibility of non-parametric methods. Kernel-based methods estimate the density distribution of the input space and identify outliers as lying in regions of low density. Roberts and Tarassenko (1995) and Bishop (1994) use Gaussian mixture models to learn a model of normal data by incrementally learning new exemplars. The GMM is represented by equation (7):

$$p(\mathbf{t}|\mathbf{x}) = \sum_{j=1}^{M} \alpha_j(\mathbf{x})\phi_j(\mathbf{t}|\mathbf{x}) \tag{7}$$

where M is the number of kernels (ϕ), $\alpha_j(\mathbf{x})$ the mixing coefficients, $\mathbf{x}$ the input vector and $\mathbf{t}$ the target vector. Tarassenko and Roberts classify EEG signatures to detect abnormal signals which represent medical conditions such as epilepsy. In both approaches, each mixture represents a kernel whose width is autonomously determined by the spread of the data. In Bishop's approach the number of mixture models is determined using cross-validation. Tarassenko and Roberts' technique adds new mixture models incrementally. If the mixture that best represents the new exemplar is above a threshold distance, then the algorithm adds a new mixture. This distance threshold is determined autonomously during system training. Once training is completed, the final distance threshold represents the novelty threshold for new items to compare against. A Gaussian probability density function is defined by equation (8):

$$\phi_j(\mathbf{t}|\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}\sigma_j^d(\mathbf{x})} \exp\left\{ - \frac{\|\mathbf{t} - \mu_j(\mathbf{x})\|^2}{2\sigma_j^2(\mathbf{x})} \right\} \tag{8}$$

where d is the dimensionality of the input space, σ is the smoothing parameter, $\mu_j(\mathbf{x})$ represents the centre of the jth kernel and $\sigma_j^2(\mathbf{x})$ is the variance (width). This growing approach is somewhat analogous to growing neural networks in section 3 as it adds a new mixture where it is not modelling the data distribution well.

Roberts (1998) introduced extreme value theory which uses a Gaussian mixture model to represent the data distribution for outlier detection as outliers (extreme values) occur in the tails of the distributions as points V, W, X, Y and Z in Figure 2 are extreme values. Least Squares regression described earlier compares the outliers against the correlation of the data distribution whereas EVT compares them against a model of the distribution. EVT again uses EM to estimate the algorithm's parameter set. EVT examines the distribution tails and estimates the probability that a given instance is an extreme value in the distribution model given by equation (9):

$$P(\text{extreme}_x) = \exp\left\{ - \exp\left(- \frac{\mathbf{x}_m - \mu_m}{\sigma_m} \right) \right\} \tag{9}$$

The approach is more principled than the proximity-based thresholding techniques discussed previously where a point is an outlier if it exceeds a threshold distance from the normal class as the EVT threshold is set using a heuristic approach. EVT is ideal for novelty recognition where abnormal samples are difficult or costly to obtain such as rare medical cases or expensive machinery malfunctions. It is also not limited to previously seen classes and is suitable for all three types of outlier detection. A classifier, such as the multi-layer perceptron (section 3.1) or decision trees (section 4), would attempt to place a new exemplar from a previously unseen class in one of the classes it has previously learned and would fail to detect the novelty.

The regression techniques and PCA are linear models which are too simple for many practical applications. Tax et al. (1999) and DeCoste and Levine (2000) use support vector machines (SVMs) for type 2 classification which use linear models to implement complex class boundaries. They project the input data onto higher dimensional kernels using a kernel function in an attempt to find a hyper-plane that separates normal and abnormal data. Such a class-defining hyper-plane may not be evident at the lower dimensions. The kernel functions range from linear dot product, polynomial non-linear used by DeCoste and Levine, a sigmoid kernel function which is equivalent to a multi-layer perceptron see section 3.1. with no hidden layers and a Gaussian function as in equation (8) used in Tax et al. which is equivalent to a radial basis function neural network described in section 3.1. Support vectors functions are positive in the dense regions of the data distribution and negative in the sparsest regions of the distribution where the outliers lie. A support vector function is defined by equation (10):

$$\mathrm{SV} = \mathrm{sign}\left(\sum_{j=1}^{n} \alpha_j L_j K(\mathbf{x}_j, \mathbf{z}) + b \right) \tag{10}$$

where K is the Kernel function, *sign* is a function returning $+1$ if the data is positive and -1 if the data is negative, L_j is the class label, b is the bias, $\mathbf{z}$ the test input and $\mathbf{x}_j$ the trained input. The data points that define the class boundary of normality are the support vectors. Only this small set of support vectors need be stored often less than 10% of the training set so a large data set can effectively be stored using a small number of exemplars.

Tax et al. (1999) use support vector novelty detection for machine condition monitoring and medical classification. SVMs can induce a classifier from a poorly balanced data set where the abnormal/normal exemplars are disproportional which is particularly true in medical domains where abnormal or in some case normal data is difficult and costly to obtain. However, SVMs are computationally complex to determine so heuristics have been devised to prevent this (DeCoste and Levine, 2000). DeCoste and Levine adapt the conventional SVM for space instrument event detection by adapting the feature weights and the cost of false positives as their data set is predominantly negative with few positive instances of an event available for training.

3. Neural Networks

Neural network approaches are generally non-parametric and model-based, they generalise well to unseen patterns and are capable of learning complex class boundaries. After training the neural network forms a classifier. However, the entire data set has to be traversed numerous times to allow the network to settle and model the data correctly. They also require both training and testing to fine tune the network and determine threshold settings before they are ready for the classification of new data. Many neural networks are susceptible to the curse of dimensionality though less so than the statistical techniques. The neural networks attempt to fit a surface over the data and there must be sufficient data density to discern the surface. Most neural networks automatically reduce the input features to focus on the key attributes. But nevertheless, they still benefit from feature selection or lower dimensionality data projections.

3.1. *Supervised neural methods*

Supervised neural networks use the classification of the data to drive the learning process. The neural network uses the class to adjust the weights and thresholds to ensure the network can correctly classify the input. The input data is effectively modelled by the whole network with each point distributed across all nodes and the output representing the classification. For example, the data in Figure 4 is represented by the weights and connections of the entire network.

Some supervised neural networks such as the multi-layer perceptron interpolate well but perform poorly for extrapolation so cannot classify unseen instances outside the bounds of the training set. Nairac et al. (1999) and Bishop (1994) both exploit this for identifying novelties. Nairac identifies novelties in time-series data for fault diagnosis in vibration signatures of aircraft engines and Bishop monitors processes such as oil pipeline flows. The MLP is a feed forward network. Bishop and Nairac use a MLP with a single hidden layer. This allows the network to detect arbitrarily complex class boundaries. The nodes in the first layer define convex hulls analogous to the convex hull statistical methods discussed in the previous section. These then form the inputs for the second layer units which combine hulls to form complex classes. Bishop notes that the non-linearity offered by the MLP classifier provides a performance improvement compared to a linear technique. It is trained by minimising the square error between the actual value and the MLP output value given by equation (11):

$$\begin{aligned}\text{Error} = &\sum_{j=1}^{m} \int [y_j(\mathbf{x};\mathbf{w}) - \langle t_j|\mathbf{x}\rangle]^2 p(\mathbf{x})\mathrm{d}\mathbf{x} \\ &+ \sum_{j=1}^{m} \int \{\langle t_j^2|\mathbf{x}\rangle - \langle t_j|\mathbf{x}\rangle^2\} p(\mathbf{x})\mathrm{d}\mathbf{x} \qquad (11)\end{aligned}$$

from Bishop (1994) where t_j is the target class, y_j is the actual class, $p(\mathbf{x})$ is the unconditional probability density which may be estimated by, for example, Parzen windows (Bishop, 1994) (discussed in section 5) and $y_j(\mathbf{x};\mathbf{w})$ is the function mapping. Provided the function mapping is flexible, if the network has sufficient hidden units, then the minimum occurs when $y_j(\mathbf{x};\mathbf{w}) = p\langle t_j|\mathbf{x}\rangle$. The outputs of the network are the regression of the target data conditioned with the input vector. The aim is thus to approximate the regression by minimising the sum-of-squares error using a finite training set. The approximation is highest where the density of $p(\mathbf{x})$ is highest as this is where the error function penalises the network mapping if it differs from the regression. Where the density is low there is little penalisation. Bishop assumes that the MLP ‘knows’ the full scope of normality after training so the density $p(\mathbf{x})$ is high, the network is interpolating and confidence is high (where confidence is given by equation (12):

$$\sigma_y(\mathbf{x}) = \{p(\mathbf{x})\}^{\frac{1}{2}} \tag{12}$$

If a new input lies outside the trained distribution where the density $p(\mathbf{x})$ is low and the network is extrapolating, the MLP will not recognise it and the confidence is low. Nairac uses the MLP to predict the expected next value based on the previous n values. This assumes that the next value is dependent only on the previous n values and ignores extraneous factors. If the actual value differs markedly from the predicted value then the system alerts the human monitoring it.

Japkowicz et al. (1995) use an auto-associative neural network for type 3 novelty recognition. The auto associator network is also a feed-forward perceptron-based network which uses supervised learning. Auto associators decrease the number of hidden nodes during network training to induce a bottleneck. The bottleneck reduces redundancies by focusing on the key attributes while maintaining and separating the essential input information and is analogous to principal component analysis. Japkowicz trains the auto-associator with normal data only. After training, the output nodes recreate the new exemplars applied to the network as inputs. The network will successfully recreate normal data but will generate a high re-creation error for novel data. Japkowicz counts the number of outputs different from the input. If this value exceeds a pre-specified threshold then the input is recognised as novel. Japkowicz sets the threshold at a level that minimises both false positive and false negative classifications. Taylor and Addison (2000) demonstrated that auto-associative neural networks were more accurate than

SOMs or Parzen windows for novelty detection. However, auto-associators suffer slow training as with the MLP and have various data-specific parameters which must be set through empirical testing and refinement.

Hopfield networks are also auto-associative and perform supervised learning. However, all node weights are +1 or −1 and it is fully connected with no bottleneck. Hopfield nets are computationally efficient and their performance is evident when a large number of patterns is stored and the input vectors are high dimensional. Crook and Hayes (1995) use a Hopfield network for type 3 novelty detection in mobile robots. The authors train the Hopfield network with pre-classified normal exemplars. The input pattern is applied to all nodes simultaneously unlike the MLP where the input ripples through the layers of the network. The output is fed back into the network as input to all nodes. Retrieval from a Hopfield network requires several network cycles while node weights update and to allow the network to settle and converge towards the 'best compromise solution' (Beale and Jackson, 1990). However, the energy calculation requires only the calculation of the sum of the weights see equation (13):

$$\text{Energy} = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{x}_i\mathbf{x}_j\mathbf{w}_{ij} \tag{13}$$

where N is the number of neurons and thus has a fixed execution time regardless of the number of input patterns stored providing a computationally efficient retrieval mechanism. If the energy is above a threshold value calculated from the distribution model and the number of patterns stored then Crook and Hayes classify the input as novel. During empirical testing, the Hopfield network trained faster than Marsland's HSOM but the HSOM discerned finer-grained image details than the Hopfield network.

The MLP uses hyper-planes to classify the data. In contrast, the supervised radial basis function (RBF) neural network in Bishop (1994) and Brotherton et al. (1998) uses hyper-ellipsoids defined by equation (14) for k dimensional inputs:

$$s_k = \sum_{j=1}^{m}\mathbf{w}_{jk}\phi(\|\mathbf{x}-\mathbf{y}_j\|) \tag{14}$$

where $\mathbf{w}_{jk}$ is the weight from the kernel to the output, ϕ is a Gaussian function (see equation (8)) and $\|\cdots\|$ the Euclidean distance. RBF is a linear combination of basis functions similar to a Gaussian mixture model and guaranteed to produce a solution as hyper-ellipsoids guarantee linear separability (Bishop, 1995). Training the network is much faster than an MLP and requires two stages: the first stage clusters the input data into the hidden layer nodes using vector quantisation. The RBF derives the radius of the kernel from the data as with the Gaussian mixture model approach (see equation (8)) using the kernel centre and variance. The second training stage weights the hidden node outputs using least mean squares weighting to produce the required output classification from the network.

Brotherton et al. (1998) revise the network topology and vector quantisation so there is a group of hidden nodes exclusively assigned for each class with a separate vector quantisation metric for each class. This means that the groups of hidden nodes represent distinct categories unlike a conventional RBF where weight combinations of all hidden nodes represent the categories. The output from each group of nodes corresponds to the probability that the input vector belongs to the class represented by the group. Brotherton uses the adapted RBF for type 3 novelty recognition in electromagnetic data or machine vibration analysis. Brotherton's adaptation creates an incremental RBF. New nodes can be added to represent new classes just as new mixtures are added to Roberts and Tarassenko's (1995) adaptive mixture model or new category nodes are added to evolutionary neural networks.

3.2. *Unsupervised neural methods*

Supervised networks require a pre-classified data set to permit learning. If this pre-classification is unavailable then an unsupervised neural network is desirable. Unsupervised neural networks contain nodes which compete to represent portions of the data set. As with perceptron-based neural networks, decision trees or k-means, they require a training data set to allow the network to learn. They autonomously cluster the input vectors through node placement to allow the underlying data distribution to be modelled and the normal/abnormal classes differentiated. They assume that related vectors have common feature values and rely on identifying these features and their values to topologically model the data distribution.

Self organising maps, Kohonen (1997) are competitive, unsupervised neural networks. SOMs perform vector quantisation and non-linear mapping to project the data distribution onto a lower dimensional grid network whose topology needs to be pre-specified by the user. Each node in the grid has an associated weight vector analogous to the mean vector representing each cluster in a *k*-means system. The network learns by iteratively reading each input from the training data set, finding the best matching unit, updating the winner's weight vector to reflect the new match like *k*-means. However, the SOM also updates the neighbouring nodes around the winner unlike *k*-means. This clusters the network into regions of similarity. The radius of the local neighbourhood shrinks during training in the standard SOM and training terminates when the radius of the neighbourhood reaches its minimum permissible value. Conversely, Ypma and Duin (1997) enlarges the final neighbourhood to increase the SOMs 'stiffness' by providing broader nodes to prevent over fitting. After training, the network has evolved to represent the normal class. In Figure 5, the SOM nodes would model the normal class, distributed according to the probability density of the data distribution.

During novelty recognition, the unseen exemplar forms the input to the network and the SOM algorithm determines the best matching unit. In Saunders and Gero (2001a) and Vesanto et al. (1998), if the vector distance or quantisation error between the best matching unit (bmu) and new exemplar exceeds some pre-specified threshold (*d*) then the exemplar is classified as novel. Equation (15) gives the minimum vector distance for the bmu and compares this to the threshold.

$$\min\left(\sum_{i=0}^{n-1}(\mathbf{x}_i(t) - \mathbf{w}_{ij}(t))^2\right) > d \tag{15}$$

For example, if the SOM nodes modelled Figure 5 (the normal class from Figure 2), the points V, W, X, Y and Z from Figure 2 would all be distant from their respective best matching units and would exceed *d* so V, W, X, Y and Z would be identified as novel. This is analogous to a *k*-means approach (section 2.1); the SOM equates to a *k*-means clustering with *k* equivalent to the number of SOM nodes. However, SOMs use a user-specified global distance threshold whereas *k*-means autonomously determines the boundary during training allowing local setting of the radius of normality as defined by each cluster. This distance match can be over simplistic and requires the selection of a suitable distance threshold so other approaches examine the response of all nodes in the grid. Himberg et al. (2001) look at the quantisation error from the input to all nodes, given by equation (16):

$$g(\mathbf{x}, \mathbf{m}_i) = \frac{1}{1 + \left(\frac{\|\mathbf{x}-\mathbf{m}_i\|}{a}\right)^2} \tag{16}$$

where a is the average distance between each training data and its bmu $\mathbf{m}_i$. The scaling factor compares the distance between $\mathbf{x}$ and its bmu $\mathbf{m}_i$ against the overall average accuracy of the map nodes during training. Ypma and Duin (1997) uses a further advancement for mechanical fault detection. They recommend an approach unifying normalised distance measure for 2 and 3 nearest neighbours rather than a single nearest neighbour of Saunders and Gero and Vesanto and normalised mean-square error of mapping.

Saunders and Gero (2001b) and Marsland's (2001) systems use an extended SOM, the Habituating SOMs (HSOMs) where every node is connected to an output neuron through habituating synapses which perform inverse Hebbian learning by reducing their connection strength through frequency of activation. The SOM forms a novelty detector where the novelty of the input is proportional to the strength of the output thus simplifying novelty detection compared to a conventional SOM as the output node indicates novelty, there is no requirement for identifying best matching units or complex quantisation error calculations. The HSOM requires minimal computational resources and is capable of operating and performing novelty detection embedded in a robot (Marsland, 2001). It also trains quickly and accurately. However, HSOMs are susceptible to saturation where any input will produce a low strength output as the HSOM cannot add new nodes and the number of classes (groups of inputs mapping to particular nodes) in the input distribution is equal to or greater than the number of nodes. If the number of classes is known in advance, then a suitable number of nodes can be pre-selected but for many outlier detection problems, particularly on-line learning problems, this information may not be available.

Evolutionary neural network growth is analogous to Roberts and Tarassenko's (1995) approach of adding new Gaussian mixtures when a new datum does not fit any of the existing mixtures. Marsland (2001) introduces the grow when required (GWR) evolutionary neural network for type 3 novel feature detection in a mobile inspection robot (robot neotaxis). It is suited to on-line learning and novelty recognition as the network can adapt and accurately model a dynamic data distribution. The growing neural network connects the best matching node and the second best match and adds new nodes when the response of the best matching node for an input vector is below some threshold value. The GWR network grows rapidly during the first phase of training but once all inputs have been processed (assuming a static training set), then the number of nodes stabilises and the network accurately represents the data distribution. As with the SOM, the GWR would learn the normal class in Figure 5 and the points V, W, X, Y and Z from Figure 2 would all then be distant from their respective best matching units which are all within the normal class.

Caudell and Newman (1993) introduce a type 3 recogniser for time-series monitoring based on the adaptive resonance theory (ART) (Carpenter and Grossberg, 1987) incremental unsupervised neural network. The network is plastic while learning, then stable while classifying but can return to plasticity to learn again making them ideal for time-series monitoring. The ART network trains by taking a new instance vector as input, matching this against the classes currently covered by the network and if the new data does not match an existing class given by equation (17):

$$\frac{\sum_{i=1}^{n} \mathbf{w}_{ij}(t)\mathbf{x}_i}{\sum_{i=1}^{n} \mathbf{x}_i} > \rho \tag{17}$$

where n is the input dimensionality and ρ is a user-specified parameter (vigilance threshold) then it creates a new class by adding a new node within the network to accommodate it. Caudell and Newman monitor the current ART classes and when the network adds a new class this indicates a change in the time series. GWR grows similarly but GWR is topology preserving with neighbourhoods of node representing similar inputs whereas ART simply aims to cover the entire input space uniformly. By only adding a new node if the existing mapping is insufficient and leaving the network unchanged otherwise, the ART network does not suffer from over-fitting which is a problem for MLPs and decision trees. The training phase also requires only a single pass through the input vectors unlike the previously described neural networks which require repeated presentation of the input vectors to train. During empirical evaluation, Marsland (2001) noted that the performance of the ART network was extremely disappointing for robot neotaxis. The ART network is not robust and thus susceptible to noise. If the vigilance parameter is set too high the network becomes too sensitive and adds new classes too often.

4. Machine Learning

Much outlier detection has only focused on continuous real-valued data attributes there has been little focus on categorical data. Most statistical and neural approaches require cardinal or at the least ordinal data to allow vector distances to be calculated and have no mechanism for processing categorical data with no implicit ordering. John (1995) and Skalak and Rissland (1990) use a C4.5 decision tree to detect outliers in categorical data and thus identify errors and unexpected entries in databases. Decision trees do not require any prior knowledge of the data unlike many statistical methods or neural methods that require parameters or distribution models derived from the data set. Conversely, decision trees have simple class boundaries compared with the complex class boundaries yielded by neural or SVM approaches. Decision trees are robust, do not suffer the curse of dimensionality as they focus on the salient attributes, and work well on noisy data. C4.5 is scalable and can accommodate large data sets and high dimensional data with acceptable running times as they only require a single training phase. C4.5 is suited to type 2 classifier systems but not to novelty recognisers which require algorithms such as *k*-means which can define a boundary of normality and recognise whether new data lies inside or outside the boundary. However, decision trees are dependent on the coverage of the training data as with many classifiers. They are also susceptible to over fitting where they do not generalise well to completely novel instances which is also a problem for many neural network methods. There are two solutions to over-fitting: pre-selection of records or pruning.

Skalak and Rissland (1990) pre-select cases using the taxonomy from a case-based retrieval algorithm to pinpoint the 'most-on-point-cases' and exclude outliers. They then train a decision tree with this pre-selected subset of normal cases. Many authors recommend pruning superfluous nodes to improve the generalisation capabilities and prevent over-fitting. John (1995) exploits this tactic and uses repeated pruning and retraining of the decision tree to derive the optimal tree representation until no further pruning is possible. The pruned nodes represent the outliers in the database and are systematically removed until only the majority of normal points are left.

Arning et al. (1996) also uses pruning but in a set-based machine-learning approach. It processes categorical data and can identify outliers where there are errors in any combination of variables. Arning identifies the subset of the data to discard that produces the greatest reduction in complexity relative to the amount of data discarded. He examines the data as a sequence and uses a set-based dissimilarity function to determine how dissimilar the new exemplar is compared to the set of exemplars already examined. A large dissimilarity indicates a potential outlier. The algorithm can run in linear time though this relies on prior knowledge of the data but even without prior knowledge, it is still feasible to process large data mining data sets with the approach. However, even the authors note that it is not possible to have a universally applicable dissimilarity function as the function accuracy is data dependent. This pruning is analogous to the statistical techniques (see section 2.2) of convex peeling or Torr and Murray's approach of shedding points and re-fitting the regression line to the remaining data.

Another machine learning technique exploited for outlier detection is rule-based systems which are very similar to decision trees as they both test a series of conditions (antecedents) before producing a conclusion (class). In fact rules may be generated directly from the paths in the decision tree. Rule-based systems are more flexible and incremental than decision trees as new rules may be added or rules amended without disturbing the existing rules. A decision tree may require the generation of a complete new tree. Fawcett and Provost (1999) describe their DC-1 activity monitoring system for detecting fraudulent activity or news story monitoring. The rule-based module may be either a classifier learning classification rules from both normal and abnormal training data or a recogniser trained on normal data only and learning rules to pinpoint changes that identify fraudulent activity. The learned rules create profiling monitors for each rule modelling the behaviour of a single entity, such as a phone account. When the activity of the entity deviates from the expected, the output from the profiling monitor reflects this. All profiling monitors relating to a single entity feed into a detector which combines the inputs and generates an alarm if the deviation exceeds a threshold. This approach detects deviations in sequences compared to Nairac's time-series approach which uses one-step ahead prediction to compare the expected next value in a sequence with the actual value measured. There is no prediction in activity monitoring all comparisons use actual measured values.

Lane and Brodley (1997a, b) introduced a similar approach for activity monitoring using similarity-based matching. The system stores and records the activities of users on a computer system as sequences. The system uses similarity-based matching to compare command sequences issued by users against the stored sequences. Due to the asynchronous nature of human-computer interaction, a new fraudulent command sequence is unlikely to exactly match a previously stored sequence. The matching metric must allow for the interpolation of gaps and non-matching sub-sequences reflecting, for example, the user collecting an incoming e-mail. Lane and Brodley take their cue from Michalski's (Dietterich and Michalski, 1985) sequential data modelling techniques and generate a validity analysis for each step in the command sequence entered. They incorporate adjacency into their similarity metric. If the similarity value is between a lower and upper bound threshold then the current sequence up to and including the current command is not anomalous.

Data mining algorithms such as BIRCH (Zhang et al., 1996) and DBSCAN (Ester et al., 1996) are robust and as such tolerant to outliers but were specifically optimised for clustering large data sets. They are based around tree structured indices and cluster hierarchies. Both techniques are capable of identifying outliers but cannot provide a degree of novelty score. BIRCH (Zhang et al., 1996) uses local clustering and data compression and has O(n) running time. BIRCH can operate incrementally, clustering data as and when it becomes available. However, it is limited to numerical data and is dependent on the order of presentation of the data set. It uses a specialised balanced tree structure called a CF-tree which is designed for large, high dimensional data sets and induces a good quality clustering from only a single pass through the data set which can optionally be improved by further passes. It can only store a limited number of records in each tree node and this can lead to an artificial cluster topology.

DBSCAN (Ester et al., 1996) is density-based with O(n log n) running time and generates an R*-tree to cluster the data and identify the k nearest neighbours. It can identify clusters of arbitrary shape like the Gaussian mixture model described earlier. DBSCAN has two user-specified parameters which determine the density of the data in the tree but it autonomously determines the number of clusters unlike for example the k-means algorithm. One parameter, the number of neighbours is generally set to 4 but the other parameter (the neighbourhood distance) requires O(n log n) time to calculate. The pre-initialisation step for DBSCAN calculates the distance between a point and its fourth nearest neighbours, plots the distance graph and then requires the user to find a valley in the distance plot. The user must identify where the valley begins from the graph profile and set the parameter accordingly.

5. Hybrid Systems

The most recent development in outlier detection technology is hybrid systems. The hybrid systems discussed in this section incorporate algorithms from at least two of the preceding sections (statistical, neural or machine learning methods). Hybridisation is used variously to overcome deficiencies with one particular classification algorithm, to exploit the advantages of multiple approaches while overcoming their weaknesses or using a meta-classifier to reconcile the outputs from multiple classifiers to handle all situations. We describe approaches where an additional algorithm is incorporated to overcome weaknesses with the primary algorithm next.

The MLP described in section 3 interpolates well but cannot extrapolate to unseen instances. Therefore, Bishop (1994) augments his MLP with a Parzen window novelty recogniser. The MLP classifies new items similar to those already seen with high confidence. The Parzen window provides an estimate of the probability density to induce a confidence estimate. Any completely novel instances will lie in a low density and the MLP prediction confidence will be low. Parzen windows are kernel-based algorithms which use various kernel functions to represent the data. Bishop uses one Gaussian kernel (as defined by equation (8)) per input vector with the kernel centred on the attribute values of the vector. The Parzen window method can provide a hard boundary for outlier detection by classifying new exemplar as normal if they belong to a kernel and an outlier otherwise. It can also provide a soft boundary by calculating the probability that the new exemplar belongs to a Gaussian kernel. Nairac et al. (1999) similarly incorporate a k-means module with the MLP module in an aircraft engine fault diagnosis system. The k-means module partitions the data and models graph shape normality. It can detect anomalies in the overall vibration shape of new signatures and the MLP detects transitions within the vibration graphs through one-step ahead prediction described in section 3.

Smyth (1994) stabilises the output from an MLP to monitor space probe data for fault detection using a hidden Markov model (HMM) as the MLP is susceptible to rapid prediction changes many of which are extraneous. The approach uses time-series data, a snapshot of the space probe transmitted periodically with equal intervals between transmissions. The feed-forward MLP trained using back propagation aims to model the probe and predicts the next state of the probe. However, the MLP has a tendency to switch between states at an artificially high rate so Smyth uses a HMM to correlate the state estimations and smooth the outputs producing a predictor with high accuracy. Hollmen and Tresp (1999) incorporate hierarchical HMMs with EM which helps determine the parameters for time series analysis in the task of cell-phone fraud detection. The authors use the HMM to predict one-step ahead values for a fraud variable in the time series given the current and previous states of a user's account and the EM algorithm is used to provide optimal parameter settings for the HMM to enable accurate one-step ahead prediction.

Hickinbotham and Austin (2000) introduced a strain gauge fault detection system to detect gauges which are not reading correctly. The system assimilates Gaussian basis function networks (GBFN) and principal components. The technique produces a frequency of occurrence matrix (FOOM) from data taken from aircraft strain gauges operating under normal conditions. The FOOM represents the normal conditions expected during a flight with readings weighted according to their likelihood of occurrence. The approach extracts two unlikeliness features and two principal components from the FOOM and feeds them into a series of GBFN, each trained with different parameter settings. The system selects the GBFN with highest classification accuracy for the data and uses this to classify new FOOMs by thresholding for novelty. A high novelty reading indicates a strain gauge error. Hollier and Austin (2002) have incorporated a further component into the strain gauge system to detect another form of strain gauge error not covered by the previous approach. They incorporate auxiliary data comprising a vector of accelerations encountered during a flight. Hollier and Austin find the maximally correlated components between the FOOM and the mean and first principal component of the auxiliary data. By establishing a relationship between the auxiliary data and normal FOOMs, they can detect corrupted FOOMs when the relationship breaks down.

Authors are beginning to examine the possibilities of ensemble classifiers where a number of different classification algorithms are used and the results from each classifier amalgamated using a suitable metric. From the preceding discussion, it is apparent that all classifiers have differing strengths and weaknesses so a combined approach can exploit the strengths of each classifier while mitigating their weaknesses.

The JAM system (Java agents for meta-learning) being developed at Columbia University (Stolfo et al., 1997) incorporates five machine-learning techniques into a single modular architecture. JAM uses meta-learning to assimilate the results from the various modules and produce a single system output. JAM modules are based on the ID3 decision tree (Quinlan, 1986), its successors CART (Breiman et al., 1984) and C4.5 (Quinlan, 1993), Ripper (a rule based learner; Cohen, 1995) and a naive Bayes classifier. In experiments, Prodromidis and Stolfo (1998) demonstrated that these classifiers were complimentary, each performing well on some data sets and comparatively poorly on other data sets. By identifying which approaches excel on which data sets and under which conditions the meta-learner can assess the degree of accuracy and confidence for the outputs from the various classifiers and select the appropriate technique for the data. Ensemble classifiers also permit sub-division of the training data with individual classifier modules trained on subsets of the data. By using multiple data models, JAM mitigates for any loss of accuracy in any one model due to data subdivision. It allows parallel training and classification with each module classifying a new instance in parallel and the meta-learner metric assimilating the results.

Brodley and Friedl (1996) also use an ensemble approach for identifying misclassified instances in a training set of pixels from satellite images of land where each pixel needs to be classified as for example, grassland, woodland, etc. They use three classifiers: a decision tree, a nearest neighbour and a linear machine. The system combines the outputs from the three classifiers using consensus voting. Consensus voting is a more conservative approach than the widely used majority voting. In consensus voting, the system only eliminates an instance from the data set if all three classifiers identify it as an outlier whereas a majority voting system eliminates an instance if the majority of classifiers identify it as an outlier. Consensus voting favours false negatives over false positives with respect to identifying outliers. The three classifiers each use a different distribution model so if all three agree that a point is an outlier then the user can be confident about the decision.

Combining multiple classifiers must be done judiciously. An ensemble should obey Occam's Razor, it should be as simple as possible with minimal redundancy, as superfluous modules waste resources, increase complexity and slow processing. Paradoxically, identifying the best combination of classifiers is a combinatorial problem.

6. Conclusions

There is no single universally applicable or generic outlier detection approach. From the previous descriptions, authors have applied a wide variety of techniques covering the full gamut of statistical, neural and machine learning techniques. We have tried to provide a broad sample of current techniques but obviously, we are unable to describe all approaches in a single paper. We hope to have provided the reader with a feel of the diversity and multiplicity of techniques available.

In outlier detection, the developer should select an algorithm that is suitable for their data set in terms of the correct distribution model, the correct attribute types, the scalability, the speed, any incremental capabilities to allow new exemplars to be stored and the modelling accuracy.

The developer should also consider which of the three fundamental approaches is suitable for their problem, a clustering approach, a classification approach or a novelty approach. This will depend on: the data type, whether the data is pre-labelled, the ground truth of the labelling (i.e., whether any novel records will be mislabelled as normal), how they wish to detect outliers and how they wish to handle them.

How the developers wish to handle outliers is very important, whether they wish to expunge them from future processing in a diagnostic clustering or a recognition system or retain them with an appropriate label in a accommodating clustering or a classification system. Developers should also consider whether they wish to simply classify a new exemplar as an outlier or not or whether they wish to apply a scale of 'outlierness' such as a percentage probability that the particular point is an outlier. The developer should carefully consider any pre-processing they will perform on the data such as feature extraction or prototyping to reduce the storage overhead of the data. Alternatively they can select an approach such as *k*-means or support vector machines that store only a minimal data set which effectively covers the data distribution through a small subset of seminal exemplars. Francis et al. (1999) showed equivalent classification accuracy when the data set of a novelty detection system based on neural network approaches was pre-processed with feature extraction compared to the classification accuracy with that of the novelty detector trained using the full range of attributes. The authors used linear regression to combine multiple attributes into single dependent attributes.

Acknowledgements

We wish to thank the reviewers for their insightful comments that allowed us to improve this paper. This work was supported by EPSRC Grant No. GR/R55191/01.

Note

There are statistical models for symbolic data (such as log-linear models) but these are not generally used for outlier analysis.

References

Aggarwal, C. C. & Yu, P. S. (2001). Outlier Detection for High Dimensional Data. *Proceedings of the ACM SIGMOD Conference 2001.*

Aha, D. W. & Bankert, R. B. (1994). Feature Selection for Case-Based Classification of Cloud Types: An Empirical Comparison. *Proceedings of the AAAI-94 Workshop on Case-Based Reasoning*.

Allan, J., Carbonell, J., Doddington, G., Yamron, J. & Yang, Y. (1998). Topic Detection and Tracking Pilot Study: Final Report. *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*.

Arning, A., Agrawal, R. & Raghavan, P. (1996). A Linear Method for Deviation Detection in Large Databases. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 164–169.

Baker, L. D., Hofmann, T., McCallum, A. K. & Yang, Y. (1999). A Hierarchical Probabilistic Model for Novelty Detection in Text. *NIPS'99*, *Unpublished manuscript*.

Barnett, V. & Lewis, T. (1994). *Outliers in Statistical Data*, 3rd edn. John Wiley & Sons.

Beale, R. & Jackson, T. (1990). *Neural Computing*: *An Introduction*. Bristol, UK and Philadelphia, PA: Institute of Physics Publishing.

Bishop, C. M. (1994). Novelty detection & Neural Network validation. *Proceedings of the IEE Conference on Vision*, *Image and Signal Processing*, 217–222.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

Blake, C. L. & Merz, C. J. (1998). UCI Repository of Machine Learning Databases, http://www.ics.uci.edu/mlearn/MLRepository.html, University of California, Irvine, Department of Information and Computer Sciences.

Bolton, R. J. & Hand, D. J. (2001). Unsupervised Profiling Methods for Fraud Detection. *Credit Scoring and Credit Control VII*, *Edinburgh*, *UK*, *5–7 September*.

Bradley, P. S., Fayyad, U. M. & Mangasarian, O. L. (1999). Mathematical Programming for Data Mining: Formulations and Challenges. *INFORMS Journal on Computing* **11**(3): 217–238.

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group.

Brodley, C. E. & Friedl, M. A. (1996). Identifying and Eliminating Mislabeled Training Instances. *Proceedings of the 13th National Conference on Artificial Intelligence*, 799–805, AAAI Press.

Brotherton, T., Johnson, T. & Chadderdon, G. (1998). Classification and Novelty Detection using Linear Models and a Class Dependent – Elliptical Bassi Function Neural Network. *Proceedings of the International conference on neural networks*. Anchorage, Alaska.

Byers, S. & Raftery, A. E. (1998). Nearest Neighbor Clutter Removal for Estimating Features in Spatial Point Processes. *Journal of the American Statistical Association* **93**(442): 577–584.

Carpenter, G. & Grossberg, S. (1987). 'A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine'. *Computer Vision*, *Graphics*, *and Image Processing* **37**: 54–115.

Caudell, T. P. & Newman, D. S. (1993). An Adaptive Resonance Architecture to Define Normality and Detect Novelties in Time Series and Databases. *IEEE World Congress on Neural Networks*, *Portland*, *Oregon*. 166–176.

Cohen, W. W. (1995). Fast Effective Rule Induction. *International Conference on Machine Learning*, 115–123.

Crook, P. & Hayes, G. (1995). A Robot Implementation of a Biologically Inspired Method for Novelty Detection. *Proceedings of TIMR-2001*, *Towards Intelligent Mobile Robots* Manchester.

Dasgupta, D. & Forrest, S. (1996). Novelty Detection in Time Series Data Using Ideas from Immunology. *Proceedings of the Fifth International Conference on Intelligent Systems*.

Datta, P. & Kibler, D. (1995). Learning prototypical concept descriptions. *Proceedings of the 12th International Conference on Machine Learning*, 158–166, Morgan Kaufmann.

DeCoste, D. & Levine, M. B. (2000). Automated Event Detection in Space Instruments: A Case Study Using IPEX-2 Data and Support Vector Machines. *Proceedings of the SPIE Conference on Astronomical Telescopes and Space Instrumentation*.

Dietterich, T. G. & Michalski, R. S. (1986). Learning to Predict Sequences. In Michalski, Carbonell & Mitchell (eds.) *Machine Learning*: *An Artificial Intelligence Approach* San Mateo, CA: Morgan Kaufmann.

Ester, M., Kriegel, H. -P. & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, *Portland*, *Oregon*, 226–231. AAAI Press.

Faloutsos, C., Korn, F., Labrinidis, A., Kotidis, Y., Kaplunovich, A. & Perkovic, D. (1997). Quantifiable Data Mining Using Principal Component Analysis. Technical Report CS-TR-3754, Institute for Systems Research, University of Maryland, College Park, MD.

Fawcett, T. & Provost, F. J. (1999). Activity Monitoring: Noticing Interesting Changes in Behavior. *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 53–62.

Francis, J., Addison, D., Wermter, S. & MacIntyre, J. (1999). Effectiveness of Feature Extraction in Neural Network Architectures for Novelty Detection. *Proceedings of the ICANN Conference*.

Grubbs, F. E. (1969). Procedures for Detecting Outlying Observations in Samples. *Technometrics* **11**: 1–21.

Hickinbotham, S. & Austin, J. (2000). Novelty Detection in Airframe Strain Data. *Proceedings of 15th International Conference on Pattern Recognition*, Barcelona, 536–539.

Himberg, J., Jussi, A., Alhoniemi, E., Vesanto, J. & Simula, O. (2001). The Self-Organizing Map as a Tool in Knowledge Engineering, In *Pattern Recognition in Soft Computing Paradigm*, 38–65. Soft Computing. World Scientific Publishing.

Hollier, G. & Austin, J. (2002). Novelty Detection for Strain-Gauge Degradation Using Maximally Correlated Components. *Proceedings of the European Symposium on Artificial Neural Networks*, *ESANN'2002*, *Bruges*, 257–262.

Hollmen, J. & Tresp, V. (1999). Call-based Fraud Detection in Mobile Communication Networks using a Hierarchical Regime-Switching Model. *Advances in Neural Information Processing Systems – Proceedings of the 1998 Conference* (*NIPS'11*), 889–895, MIT Press.

Japkowicz, N., Myers, C. & Gluck M. A. (1995). A Novelty Detection Approach to Classification. *Proceedings of the 14th International Conference on Artificial Intelligence* (*IJCAI-95*), 518–523.

John, G. H. (1995). Robust Decision Trees: Removing Outliers from Databases. *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, 174–179. Menlo Park, CA: AAAI Press.

Knorr, E. M. & Ng, R. T. (1998). Algorithms for Mining Distance-Based Outliers in Large Datasets. *Proceedings of the VLDB Conference*, 392–403, New York, USA.

Kohonen, T. (1997). *Self-Organizing Maps*, Vol. 2. Springer-Verlag, Heidelberg.

Lane, T. & Brodley, C. E. (1997a). Applications of Machine Learning to Anomaly Detection. In Adey, R.A., Rzevski, G, and Teti, T. (eds.) *Applications of Artificial Intelligence in Engineering X11*, 113–14, Southampton, UK: Comput. Mech. Publications.

Lane, T. & Brodley, C. E. (1997b). Sequence matching and learning in anomaly detection for computer security. *AAAI Workshop*: *AI Approaches to Fraud Detection and Risk Management*, 43–49. AAAI Press.

Laurikkala, J., Juhola, M. & Kentala, E. (2000). Informal Identification of Outliers in Medical Data. *Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology IDAMAP-2000 Berlin*, *22 August*. *Organized as a workshop of the 14th European Conference on Artificial Intelligence ECAI-2000*.

Marsland, S. (2001). On-Line Novelty Detection Through Self-Organisation, with Application to Inspection Robotics. Ph.D. Thesis, Faculty of Science and Engineering, University of Manchester, UK.

Nairac, A., Townsend, N., Carr, R., King, S., Cowley, P. & Tarassenko, L. (1999). A System for the Analysis of Jet System Vibration Data. Integrated ComputerAided Engineering **6**(1): 53–65.

Ng, R. T. & Han, J. (1994). Efficient and Effective Clustering Methods for Spatial Data Mining. *Proceedings of the 20th International Conference on Very Large Data Bases*, *September 12–15*, *1994*, *Santiago*, *Chile*, 144–155. Morgan Kaufmann Publishers.

Parra, L., Deco, G. & Miesbach S. (1996). Statistical Independence and Novelty Detection with Information Preserving Nonlinear Maps. Neural Computation **8** (2): 260–269.

Prodromidis, A. L. & Stolfo, S. J. (1998). Mining Databases with Different Schemas: Integrating Incompatible Classifiers. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 314–318.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning* **1**(1): 81–106.

Quinlan, J. R. (1993). C4.5: *Programs for Machine Learning*. Morgan Kaufmann.

Ramaswamy, S., Rastogi, R. & Shim, K. (2000). Efficient Algorithms for Mining Outliers from Large Data Sets. *Proceedings of the ACM SIGMOD Conference on Management of Data*, *Dallas*, *TX*, 427–438.

Roberts, S. J. (1998). Novelty Detection Using Extreme Value Statistics. *IEE Proceedings on Vision*, *Image and Signal Processing* **146**(3): 124–129.

Roberts, S. & Tarassenko, L. (1995). A Probabilistic Resource Allocating Network for Novelty Detection. Neural Computation **6**: 270–284.

Rousseeuw, P. & Leroy, A. (1996). *Robust Regression and Outlier Detection*, 3rd edn. John Wiley & Sons.

Saunders, R. & Gero, J. S. (2001a). A Curious Design Agent: A Computational Model of Novelty-Seeking Behaviour in Design. *Proceedings of the Sixth Conference on Computer Aided Architectural Design Research in Asia* (*CAADRIA 2001*), *Sydney*.

Saunders, R. & Gero, J. S. (2001b). Designing for Interest and Novelty: Motivating Design Agents. *Proceedings of CAAD Futures 2001*, *Eindhoven*.

Shekhar, S., Lu, C. & Zhang, P. (2001). Detecting Graph-Based Spatial Outliers: Algorithms and Applications. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Skalak, D. B. (1994). Prototype and feature selection by sampling and random mutation hill climbing algorithms. *Machine Learning*: *Proceedings of the Eleventh International Conference*, 293–301.

Skalak, D. B. & Rissland, E. L. (1990). Inductive Learning in a Mixed Paradigm Setting. *Proceedings of the Eighth National Conference on Artificial Intelligence, Boston, MA,* 840–847.

Smyth, P. (1994). Markov Monitoring with Unknown States. *IEEE Journal on Selected Areas in Communications, Special Issue on Intelligent Signal Processing for Communications* **12**(9): 1600–1612.

Stolfo, S. J., Prodromidis, A. L., Tselepis, S., Lee, W., Fan, D. W. & Chan, P. K. (1997). JAM: Java Agents for Meta-Learning over Distributed Databases. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 74–81.

Tang, J., Chen, Z., Fu, A. & Cheung, D. (2002). A Robust Outlier Detection Scheme in Large Data Sets, 6th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Taipei, Taiwan, May, 2002.

Tax, D. M. J., Ypma, A. & Duin, R. P. W. (1999). Support Vector Data Description Applied to Machine Vibration Analysis. *Proceedings of ASCI'99*, *Heijen*, *Netherlands.*

Taylor, O. & Addison, D. (2000). Novelty Detection Using Neural Network Technology. *Proceedings of the COMADEN Conference.*

Torr, P. H. S. & Murray, D. W. (1993). Outlier Detection and Motion Segmentation. *Proceedings of SPIE.*

Vesanto, J., Himberg, J., Siponen, M. & Simula, O. (1998). Enhancing SOM Based Data Visualization. *Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems. Methodologies for the Conception, Design and Application of Soft Computing*, Vol. 1, 64–67, Singapore: World Scientific.

Wettschereck, D. (1994). A Study of Distance-based Machine Learning Algorithms. Ph.D. Thesis, Department of Computer Science, Oregon State University, Corvallis.

Ypma, A. & Duin, R. P. W. (1997). Novelty Detection Using Self-Organizing Maps. In Kasabov, N., Kozma, R. Ko, K., O'Shea, R., Coghill, G. & Gedeon, T. (eds.) *Progress in Connectionist-Based Information Systems*, Vol. 2, London: Springer. 1322–1325.

Zhang, T., Ramakrishnan, R. & Livny, M. (1996). BIRCH: An Efficient Data Clustering Method for Very Large Databases'. In Jagadish, H. V. & Mumick, I. S. (eds.) *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, *Montreal*, *Quebec*, *Canada*, June 4–6, 1996, 103–114. ACM Press.